DD2420 - Tutorial 7

Magnus Pierrau

February 2020

1 Rejection Sampling

1.1 a)

We want to show that the conditional distribution of Y, $F(y)$, can be estimated by accepting/rejecting samples from $U \leq \frac{f(Y)}{cg(Y)}$, where g is our proposal distribution, which, after multiplication by the constant c covers the support of f. We need to show that $P(Y \leq y \mid U \leq \frac{f(Y)}{cg(Y)} = F(y)$, where $F(y)$ is the cumulative density function (CMD) of f. This can be done by utilizing Bayes theorem with $A = \{Y \leq y\}$ and $B = \{U \leq \frac{f(Y)}{cg(Y)}\}$. Using Bayes theorem we have that

$$
\begin{aligned}
P\left(Y \leq y \mid U \leq \frac{f(Y)}{cg(Y)}\right) &= P\left(U \leq \frac{f(Y)}{cg(Y)} \mid Y \leq y\right)\frac{P(Y \leq y)}{1/c} \\
=&P\left(U \leq \frac{f(Y)}{cg(Y)} \mid Y \leq y\right)\frac{G(y)}{1/c}
\end{aligned} \tag{1}
$$

Here $G(y)$ is the CMD of g. We can rewrite the first part of (1) as

$$
P\left(U \leq \frac{f(Y)}{cg(Y)} \mid Y \leq y\right) = \frac{P\left(U \leq \frac{f(Y)}{cg(Y)}, Y \leq\right)}{G(y)},
$$

and then condition on apply the definition of a CMD:

$$
\begin{aligned}
\frac{P\left(U \leq \frac{f(Y)}{cg(Y)}, Y \leq y\right)}{G(y)} =& \int_{-\infty}^{y} \frac{P\left(U \leq \frac{f(Y)}{cg(Y)} \mid Y = w\right)}{G(y)} g(w)dw \\
=& \frac{1}{G(y)} \int_{-\infty}^{y} \frac{f(w)}{cg(w)} g(w)dw \\
=& \frac{1}{cG(y)} \int_{-\infty}^{y} f(w)dw \\
=& \frac{F(y)}{cG(y)}
\end{aligned} \tag{2}
$$

Where, in (2) we used that $P(U \leq X \mid X = x) = x$. This gives that (1) now becomes

$$P\left(U \leq \frac{f(Y)}{cg(Y)} \mid Y \leq y\right) \frac{G(y)}{1/c} = \frac{F(y)}{cG(y)} \frac{G(y)}{1/c} = F(y). \tag{3}$$

Q.E.D.

1.2 b)

We derive the acceptance probability $P\left(u < \frac{\tilde{p}(x)}{Mq(x)}\right)$ for the unnormalized case, where $p(x) = \frac{1}{Z}\tilde{p}(x)$. This is done with the help of conditional expectations and that for some event A it holds that $\mathbb{E}[\mathbb{1}\{P(A)\}] = P(A)$, where $\mathbb{1}$ is the indicator function.

$$\begin{aligned}
&P\left(U < \frac{\tilde{p}(X)}{Mq(X)}\right) && (4)\\
=&\mathbb{E}_q\left[\mathbb{1}\left\{U < \frac{\tilde{p}(X)}{Mq(X)}\right\}\right] && (5)\\
=&\mathbb{E}_q\left[\mathbb{E}_x\left[\mathbb{1}\left\{U < \frac{\tilde{p}(X)}{Mq(X)} \mid X = x\right\}\right]\right] \\
=&\mathbb{E}_q\left[P\left(U < \frac{\tilde{p}(X)}{Mq(X)} \mid X = x\right)\right] && (6)\\
=&\mathbb{E}_q\left[\frac{\tilde{p}(X)}{Mq(X)}\right] = \sum_{x \in \chi} \frac{1}{M} \frac{\tilde{p}(x)}{q(x)} q(x) && (7)\\
=&\frac{1}{M} \sum_{x \in \chi} \tilde{p}(x) = \frac{1}{M} \sum_{x \in \chi} Zp(x) && (8)\\
=&\frac{Z}{M} \sum_{x \in \chi} p(x) = \frac{Z}{M} && (9)
\end{aligned}$$

2 Importance Sampling

2.1 a)

We want to show that

$$\mathbb{E}_p[f(x)] \approx \frac{1}{L} \sum_{l=1}^{L} f(x^{(l)}) w(x^{(l)}), \tag{10}$$

where $x^{(l)} \sim q(x)$, which is our proposal distribution. Here, $w(x^{(l)}) := \frac{p(x^{(l)})}{q(x^{(l)})}$ is the importance weight of the lth sample $x^{(l)}$ and f is some function that we want to approximate the expectation of, with respect to some true distribution p, and then applying the law of large numbers.

This is done by applying the definition of the expectation and multiplying and dividing by $q(x)$ and then realizing that this can be reinterpreted as an expectation over q instead of p. Like so:

$$
\begin{aligned}
\mathbb{E}_p[f(x)] =& \sum_{x \in \chi} f(x)p(x) = \sum_{x \in \chi} f(x)\frac{p(x)}{q(x)}q(x) && (11)\\
=& \mathbb{E}_q\left[f(x)\frac{p(x)}{q(x)}\right] = \mathbb{E}_q[f(x)w(x)] && (12)\\
\approx& \frac{1}{L}\sum_{l=1}^{L} f(x^{(l)})w(x^{(l)}), && (13)
\end{aligned}
$$

where $x^{(l)} \sim q(x)$.

Q.E.D.

2.2 b)

Why do we need importance weights?

If we are trying to estimate the density of some distribution p(x) which has a long tail, for example an exponential function, then most of the sampled $x^{(l)}$ from our proposal distribution $q(x)$ will be rejected due to the probability of acceptance being low for a large range of x.

By weighting each $x^{(l)}$ according to the ratio of how likely it is observed under p and q respectively, the "unusual" samples will be given a higher weight than the "common" samples, leading us to accept more samples and thus increasing the convergence rate.

3 Metropolis-Hastings Sampling Exercises

Exercise 1

Here we want to show that the Markov Chain transition kernel in the Metropolis-Hastings Algorithm satisfy the detailed balance condition. I.e. that the transition kernel

$$
T\left(x^{(i+1)} \mid x^{(i)}\right) = q(x^{(i+1)} \mid x^{(i)})A(x^{(i)}, x^{(i+1)}) + \delta_{x^{(i)}}(x^{(i+1)})r(x^{(i)}), \qquad (14)
$$

with $\delta_{x^{(i)}}(x^{(x+i)})$ as the Delta Dirac function,
and $r(x^{(i)}) = \int q(x^* \mid x^{(i)})(1-A(x^{(i)}, x^{(i)}))dx^*$ and $A(x^{(i)}, x^{(i+1)}) = \min\left\{1, \frac{p(x^{(i+1)})q(x^{(i)}|x^{(i+1)})}{p(x^{(i)})q(x^{(i+1)}|x^{(i)})}\right\}$,
satisfies the relation

$$p(x^{(i+1)})T(x^{(i)} \mid x^{(i+1)}) = p(x^{(i)})T(x^{(i+1)} \mid x^{(i)}). \tag{15}$$

To do this we first note that transitioning into a new state either means that $x^{(i+1)} = x^{(i)}$ or that $x^{(i+1)} \neq x^{(i)}$. These are disjoint sets and we can thus handle them separately. We will denote the cases as case **A** and **B** respectively.

We will prove that the detailed balance equation holds in both cases, one of which will be trivial.

Case A

This case occurs either when we reject the new sample x^* and thus accepting $x^{(i)}$ as our new state $x^{(i+1)}$ or because the new sample x^* is accepted as our new state and it just so happens to be that $x^* = x^{(i)}$.

The first case (reject x^*) happens with probability

$$\int q(x^* \mid x^{(i)}) \Big(1 - A(x^*, x^{(i)})\Big)\, dx^*, \tag{16}$$

and the second (accept x^*) with probability

$$q(x^{(i)} \mid x^{(i)})A(x^{(i)}, x^{(i)}). \tag{17}$$

In either way, we get the result that

$$p(x^{(i+1)})T(x^{(i)} \mid x^{(i+1)}) = p(x^{(i)})T(x^{(i)} \mid x^{(i)}) = p(x^{(i)})T(x^{(i+1)} \mid x^{(i)}). \tag{18}$$

This shows that case **A** trivially fulfills the detailed balance equation.

Case B

Applying the expression for $T(x^{(i+1)} \mid x^{(i)})$ we get that

$$\begin{align}
T(x^{(i+1)} \mid x^{(i)}) &= \min\left\{1, \frac{p(x^{(i+1)})q(x^{(i)} \mid x^{(i+1)})}{p(x^{(i)})q(x^{(i+1)} \mid x^{(i)})}\right\} q(x^{(i+1)} \mid x^{(i)}) \tag{19}\\
&= \frac{1}{p(x^{(i)})} \min\left\{p(x^{(i)} \mid q(x^{(i+1)} \mid x^{(i)}), p(x^{(i+1)})q(x^{(i)} \mid x^{(i+1)})\right\} \tag{20}
\end{align}$$

By comparing the results of the two resulting cases from the *min* function one can convince oneself that this equality actually holds. This gives that

$$
\begin{aligned}
T(x^{(i+1)} \mid x^{(i)})p(x^{(i)}) &= \frac{p(x^{(i)})}{p(x^{(i)})} \min \left\{ p(x^{(i)}), q(x^{(i+1)} \mid x^{(i)}), p(x^{(i+1)})q(x^{(i)} \mid x^{(i+1)}) \right\} \qquad (21) \\
&= \frac{p(x^{(i+1)})}{p(x^{(i+1)})} \min \left\{ p(x^{(i+1)})q(x^{(i+1)} \mid x^{(i)}), p(x^{(i)})q(x^{(i)} \mid x^{(i+1)}) \right\} \qquad (22) \\
&= T(x^{(i)} \mid x^{(i+1)})p(x^{(i+1)}). \qquad (23)
\end{aligned}
$$

This proves that the detailed balance equation holds also for the second case. Q.E.D.

Exercise 2

Sampling from an asymmetric distribution can be justified if we have prior knowledge about the system we are inferring on, informing us of non-symmetry between states, for example if we are trying to estimate a known skewed distribution or a distribution over some variance parameter which is non-negative.

Exercise 3

For a model with PGM $p(y, x) = p(x)p(y \mid x)$, what could be a reasonable proposal distribution for the independent sampler when we want to sample from posterior $p(x \mid y)$?

3.1 Coding

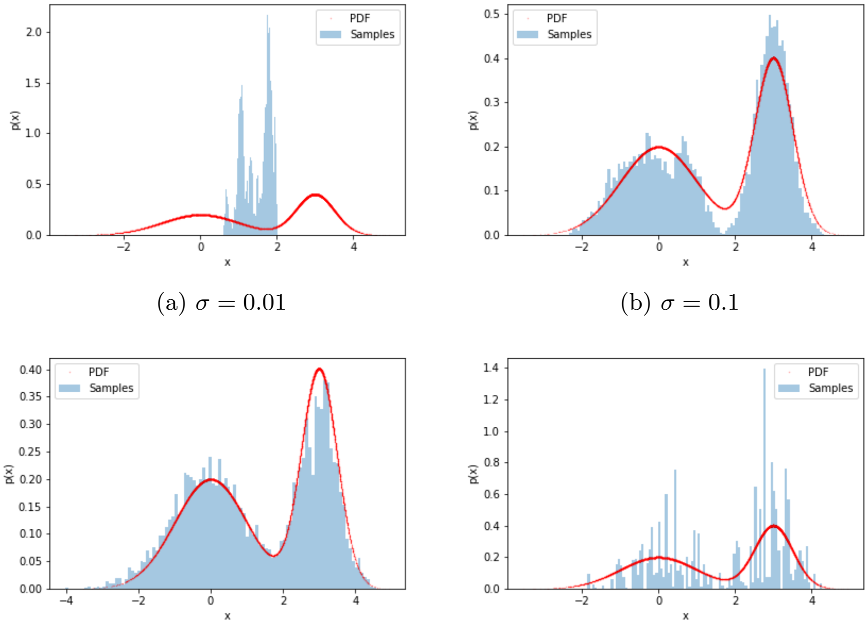
Here we implement Metropolis-Hastings algorithm from scratch in Python, allowing for the general case, when the normalizing constant of the target distribution is unknown.

As can be seen in figure 1 the results vary greatly depending on which variance is used. When the variance is too small, as in figure 1a, the samples tend to gather at the local maximum of the (unplotted) proposal distribution. The small variance does not allow the samples to wander walk too far from the previous state, as the proposal distribution is virtually zero there. This disables us from exploring the entire state space. We also note here that the estimated probability exceeding 1.0, suggesting some error in calculations.

In figure 1b the variance is increased somewhat, now allowing us to explore the entire state space. We note that the samples gather under the two modes of $p(x)$, but not traversing the space between them, where the probability is relatively low, both for the proposal distribution and the true distribution. The walk overestimates the distribution at the rightmost mode, while underestimating it between the modes. The rightmost mode is traversed frequently since it is probable to walk there under $p(x)$.

In figure 1c the variance is increased further, to 1.0. This now allows the random walk to traverse the entire state space again, and with a frequency similar to the true underlying distribution.

As we increase the variance with two magnitudes we see that this effectively serves as a uniform prior, making the proposal distribution very flat. Therefore the proposal distribution will have less impact on the walk, as $q(x^{(l)} \mid x^*)$ and $q(x^* \mid x^{(l)})$ are now very similiar. We therefore see many samples around the modes of $p(x)$. Again, the frequency exceeds 1.0. This suggests a plotting issue rather than computational error.



(a) $\sigma = 0.01$

(b) $\sigma = 0.1$

(c) $\sigma = 1.0$

(d) $\sigma = 100$

Figure 1: Random walk for Metropolis-Hastings algorithm with $N = 10000$ steps and varying standard deviation σ. The blue bars indicate the histogram over samples and the red curve the true underlying Gaussian mixture model.

4 Particle Filtering & Sequential Monte Carlo

4.1 Coding Exercise

We are given the model

$$x_t = \frac{1}{2}x_{t-1} + 25\frac{x_{t-1}}{1+x_{t-1}^2} + 8\cos(1.2t) + v_t \tag{24}$$

$$y_t = \frac{x_t^2}{20} + w_t, \tag{25}$$

where $x_1 \sim \mathcal{N}(0, \sigma_1^2 = 10)$, $v_t \sim \mathcal{N}(0, \sigma_v^2 = 10)$ and $w_t \sim \mathcal{N}(0, \sigma_w^2 = 1)$ are Gaussian noise terms.

This implies that

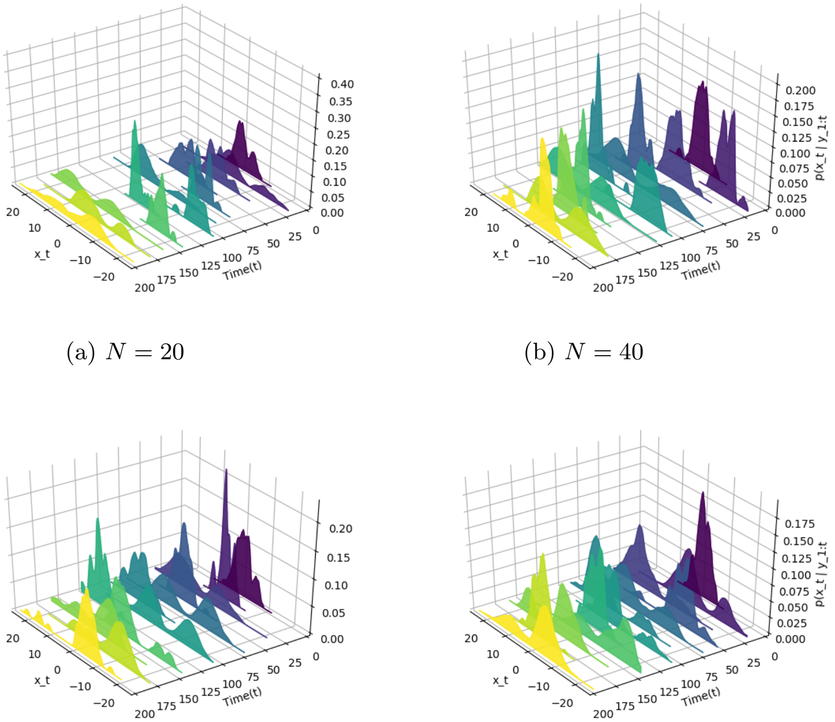
$$p(x_{(t+1)} \mid x_{(t)}) \sim \mathcal{N}(\frac{1}{2}x_{t-1} + 25\frac{x_{t-1}}{1+x_{t-1}^2} + 8\cos(1.2t), \sigma_v^2) \tag{26}$$

$$p(y_t \mid x_t) \sim \mathcal{N}(\frac{x_t^2}{20}, \sigma_w^2) \tag{27}$$

We implement the Bootstrap filter in Python and plot the estimated filtering distribution $p(x_t \mid y_{1:t})$ in figures 2 and 3 below.

By studying figures 2a - 2d we see that multimodality begins to show up strongly around $N = 40, 50$. For $N = 20$ most distributions are unimodal over most t. There are some hints of multimodality already at $N = 20$, but they are weak and only appear for larger t.

To answer when the filtering distribution does not change considerably after increasing N we did an empirical study for different N and found that when we have a $N > 1000$ there are only minor differences between the produced filtering distributions (see 3).



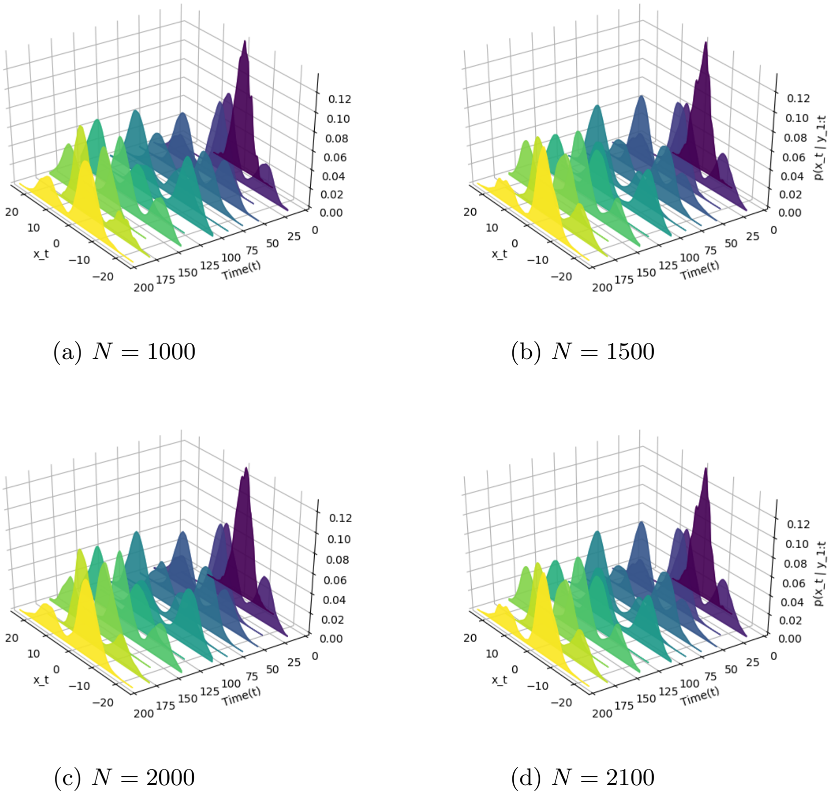
(a) $N = 20$

(b) $N = 40$

(c) $N = 50$

(d) $N = 100$

Figure 2: Filtering distribution generated by online Bootstrap particle filtering using varying number of samples N and time $T = 200$. Observe that the time steps are decreasing in the plot, from 200 at the origin to 0.



(a) $N = 1000$

(b) $N = 1500$

(c) $N = 2000$

(d) $N = 2100$

Figure 3: Filtering distribution generated by online Bootstrap particle filtering using varying number of samples N and time $T = 200$. Observe that the time steps are decreasing in the plot, from 200 at the origin to 0.